Fingerprint-based User-friendly Interface and Pocket-PID for Mobile Authentication

Kaoru Uchida
Computer and Communication Media Research, NEC Corporation
k-uchida@bc.jp.nec.com

Abstract

This paper presents an enhanced user interface, "Fingerprint User Interface (FpUI)," that exploits fingerprint identification technology, and discusses its application to information systems, specifically to a mobile authentication terminal for networked services and digital appliances. FpUI utilizes information regarding not only who put a finger on its sensor but which specific finger it was. A user can assign commands, data objects, status, or personalized settings to individual fingers for user-friendly direct manipulation. A mobile terminal, "Pocket-PID," with fast, accurate fingerprint identification capability is proposed which features (1) an easy-to-use FpUI, (2) high security (the identification function is totally enclosed within the unit, which authenticates a user's identity without the possibility of actual fingerprint data being disclosed), and (3) secure and usable non-contact communication capability which facilitates implementation of networked services based on biometric user identification.

1. Introduction

Personal identification using biometrics, i.e. personal physiological or behavioral traits, is one of the most promising "real" applications of pattern recognition; it is becoming evident that this technology has the potential to meet the increasing demand for automated user verification, and to contribute to the security of future information systems and networked services, including electronic commerce.

Fingerprint identification, the most mature biometrics technology, uses algorithms and software with proven reliability and was first used in law enforcement and later in physical access control applications. It is now also used for personal purposes, mainly for logins to standalone computers and network services [1]. In these applications, however, fingerprints have only been used as a means of personal verification, based on the fact that they are invariant with time and unique among people, and system designers have only exploited this single aspect of their potential.

We focus on the fact that a person's fingerprints are different from finger to finger as well, and propose what we call the "Fingerprint User Interface" (FpUI), which allows a user to assign commands and data objects to individual fingers to augment man-machine interactions.

In this paper, I first discuss the concept, implementation and possible applications of FpUI. Then, as a useful application for FpUI, I propose a mobile terminal, "Pocket-PID," and discuss it in detail. Finally I describe my experimental implementation of Pocket-PID.

2. Fingerprint-based user interface: FpUI

2.1. FpUI as multimodal human interface

When we interact with computer systems by, for example, hitting keys, all that the system knows is which key has been hit and when. If the keys were equipped with fingerprint sensors and software were utilized that could distinguish differences among fingerprints, a system would be able to take actions determined not only by whose but also by which finger activated a given sensor. This is the concept behind the "Fingerprint User Interface" (FpUI) for enhancing man-machine interactions [2].

An FpUI can be categorized as a multimodal user interface [3]; it introduces an individual's ten fingers as a set of new communication modalities.

Let us consider the situation in which the FpUI is applied to a single-sensor man-machine system. A table is first prepared, describing the relationship between a fingerprint and an FpUI action (i.e. what action to be taken when a person X presents his N-th finger to the sensor).

Figure 1 illustrates how FpUI works. When a user touches the sensor with a certain finger, it obtains an image of the fingerprint. Note that a good automated image capture mechanism ought to be employed so that the system will be able to obtain an image of sufficient quality when the user puts his finger naturally on the sensor. Fingerprint identification is executed on the acquired image to locate a matching fingerprint in the prepared table, and the action associated with the result is carried out in response.

0-7695-0750-6/00 $10.00 © 2000 IEEE

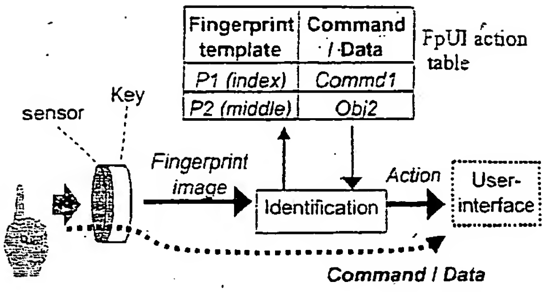

Figure 1. Fingerprint User Interface

2.2. FpUI with 1-to-N matching

To implement FpUI, we need both a small, thin fingerprint sensor and a fast, accurate fingerprint identification algorithm that can perform "1-to-N" matching.

Several solid-state fingerprint sensing chips are currently available for the fabrication of sensors that are smaller and thinner than conventional optical ones. With respect to identification, what we need is software that can search an entire fingerprint database for a match of the input image and execute the action associated with the data. This "1-to-N" matching is in contrast to the "1-to-1" matching used in most existing fingerprint-based personal verification systems, in which a user is required to claim his identity first, after which the system checks to see if the input fingerprint matches the data it holds for that identity.

2.3. Examples of FpUI applications

The FpUI concept itself is simple and straightforward, and it proves very effective in a number of ways. Below are some example of applications, that might broaden the use of biometrics technology.

2.3.1. Fingertip commands. Different commands can be assigned to different fingers. While the conventional "hitting the key" action only provides an execution trigger, this UI enables execution of specific actions tied to specific fingers. Using such "fingertip commands," an interface designer can reduce the number of required keys and avoid the use of mode keys (such as control and alt keys), which often confuse computer novices. It also facilitates the use of "blind operations," which may be needed, for example, when a car-driver operates a stereo, or when a user tries to operate appliances in darkness.

2.3.2. Fingertip saver. At the time of login, a fingerprint can be used not only for user-verification but also for system customization (e.g. desktop design, shortcuts, etc.) based on user preference; a user can also choose among multiple sets of working environments by the choice of finger used. In addition to static setups, the dynamic status of a pending session can be saved and later restored just by presenting a fingertip, so that a user can continue his work more easily.

Metaphorically, this FpUI represents "saving" a state in each finger, and its significance increases dramatically when used on a small terminal, as opposed to on a computer with a large keyboard. It is also particularly well-suited to digital appliances for children or for elderly people who are not familiar with conventional user interfaces. For example, when a group of children share a video game or an electronic picture book in home or at school, each user could make a finger "remember" how far he had gone, and could resume later with just a touch of that finger. Such "fingertip bookmarking" is a good example of the potential user-friendliness of FpUI for non-experts.

It would be particularly useful for a user to be able to resume his work, with a fingertip, at any terminal whatever among multiple client terminals connected in a network. Such a networked-FpUI could be implemented by customizing the client where the input was made according to the result of fingerprint identification carried out on the server.

2.3.3. Fingertip memo. The concept of "state memorization" suggests the idea of a user interface utilizing fingers virtually as data storage for various data objects. For example, keeping an URL in each finger would help us browse Web sites just by changing fingers. Creating documents might also be facilitated by allocating frequently used text segments (such as signatures or greetings) to fingers, so as to be able to insert them instantly. By using a "memorize then retrieve" sequence dynamically, we might copy-and-paste via multiple fingertip copy buffers (clipboards).

This application could also be used over a network, with one object that has been virtually copied to a finger on one PC being pasted on another by the touch of that finger. Figure 2 shows how this might be done. Such direct manipulation might be viewed as an extension of Pick-and-Drop [4].

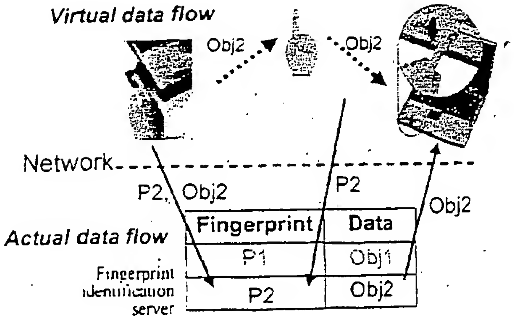

Figure 2. Fingertip memo

3. Pocket-PID for security and usability

3.1. Fingerprint identification on a mobile unit

As previously noted, FpUI shows more of its advantages on smaller terminals. The system I propose here consists of a PDA-like unit with fingerprint identification capability, "Pocket-PID" (Personal Identificator), which a user would carry, and a service terminal, typically a PC with an IrDA port, connected to a network. The Pocket-PID could be used to prove the owner's identity so that a service though the terminal could be authorized. Such a system would have the following characteristics, each of which I will discuss in the following subsections:

1. Authentication process security enclosed in the unit
2. IrDA communication for enhanced usability
3. Incorporation of FpUI

3.2. Secure management of fingerprint data

In conventional network-based services using fingerprint user verification, an authorized user's fingerprint template is stored typically either (1) in the server (as in Figure 3), or (2) in the user's IC card (as in Figure 4).

The former configuration may be sufficiently secure for intranet-based closed systems, but if we were to extend this framework to an open-network-based service, users would be likely to feel uncomfortable about the management and handling of their biometric data at the server and in the open network.

The latter design, on the other hand, provides a better solution, but even this would not be secure enough if the client machine were run by a dishonest administrator, in which case the user's fingerprint data could be secretly copied at the client terminal, and could be used for malicious purposes afterwards.

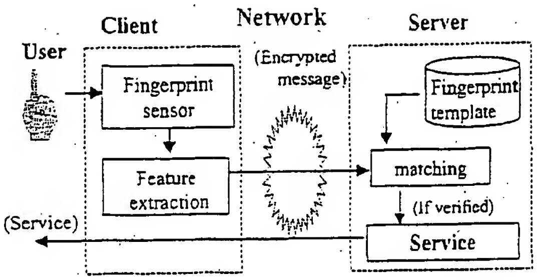

Figure 3. Server-based authentication

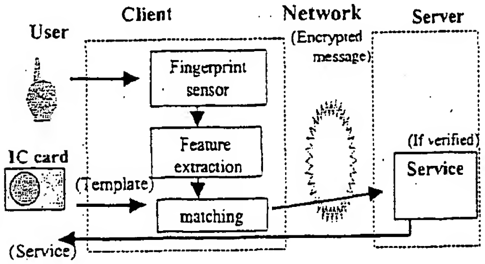

Figure 4. Authentication using an IC card

In the system I propose, on the other hand, as shown in Figure 5, the Pocket-PID unit stores the user's fingerprint template within it, has a fingerprint sensor on it, and executes verification by itself, thus avoiding the possibility of the disclosure of fingerprint information.

This all-in-one identification on a mobile unit has been achieved with the use of a thin fingerprint sensor having solid-state imaging capability and an accurate and fast identification software which runs on a low power CPU.

3.3. Usability by non-contact communication

To augment usability of the Pocket-PID, we have designed it so that it can be simply used by pointing it at a PC. It is not necessary to connect it physically by cable, the Pocket-PID communicates with PCs via a non-contact communication channel, such as IrDA or bluetooth.

But because IrDA messages can easily be intercepted and recorded, and other machines could pretend to be the user's Pocket-PID by using recorded messages to forge an illegal session, we have implemented a special protocol to prevent any such "replay attacks." In this protocol, the two parties first mutually verify that their counterpart is genuine and trustable, agree on a one-time key for encryption with a secret algorithm, and then exchange encrypted and digitally-signed messages, including fingerprint identification results.

This IrDA-based configuration also provides a solution to another possible difficulty that could hinder the wide-ranged use of fingerprints in net-based services: the cost of building and running systems. To make fingerprint-based authentication universally available (using a framework as in Figure 3 or 4) without a tool like Pocket-PID, numerous fingerprint sensors would have to be installed at various service points, including homes, retail stores, and on public service terminals. Also, the need to perform identification and to exchange fingerprint data over networks securely without the fear of jeopardizing privacy would greatly increase the software and operating costs of the whole complex system.

With the Pocket-PID configuration, on the other hand, all that a service client terminal would need is a communication port, such as an IrDA port, and communication software for transmitting the authentication interchange. Because the user's Pocket-PID unit exchanges the encrypted authentication results directly with the server, client installation and operational costs would be minimal.

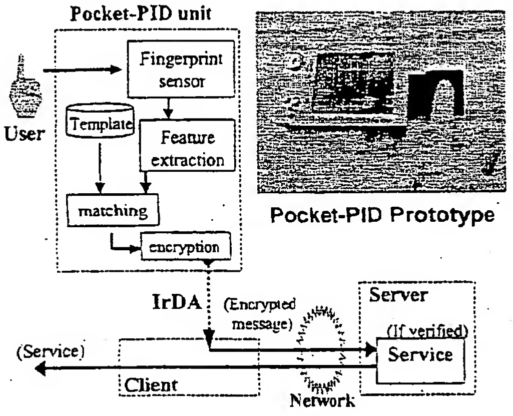

Figure 5. Authentication using Pocket-PID

3.5. FpUI on a Pocket-PID used as a digital assistant

A user of a Pocket-PID as a digital assistant would further benefit from the FpUI as well with its fingerprint identification capability.

3.5.1. Fingertip remote control. The combination of "fingertip commands" and IrDA (or bluetooth) communication makes possible "fingertip remote control," for controlling PCs or digital appliances by changing fingers. The user could register in advance relationships between individual fingers and specific infrared (or wireless) messages, and the Pocket-PID would transmit the appropriate message when a given finger was presented.

3.5.2. Typing assistant. Text-writing on a PDA might be facilitated by taking advantage of the "fingertip memo" function, with, for example, inserting frequently used phrases or a signature being inserted in response to a finger touch. In editing, a user might copy-and-paste text segments or objects by means of a fingertip clipboard.

3.5.3. Virtual data carrier. Since a fingerprint is universally unique, a public information system might be designed so that "fingertip memos" were available over the Internet. When a user wanted to take home a copy of data he had found on a public information terminal, for example, he could point his Pocket-PID at that terminal and virtually copy the data into his fingertip. He could later retrieve the data at home from a PC connected to the Internet, since the Pocket-PID holds a pointer to the actual data associated with a given finger.

3.5.4. User specific personalization. Though all these examples of FpUI usage take advantage of the differences among the fingers of a single user, naturally they all are executable only by that user since fingers are universally unique as well. This contributes greatly to privacy management and to unit personalization, and facts regarding what actions are to be taken on what kind of data will be securely protected.

4. Prototype implementation of Pocket-PID

4.1. Experimental implementation

I have designed and implemented a prototype system, based on the proposed concept, consisting of a palm-size PC as the Pocket-PID unit and a PC used as a client terminal (Figure 5).

Fingerprint identification software based on a minutia-relation algorithm [5] is run on this unit. The algorithm is basically the same one that NEC has used in a number of their systems, including AFISs (Automated Fingerprint Identification Systems) for law enforcement applications, and the software attains highly accurate 1-to-N identification at a considerable speed even on the relatively slow CPU in the unit. I have also engineered a small, thin input scanner using a capacitive fingerprint sensor chip (FPS110 from Veridicom [6]) that works on a palm-size PC.

In the prototype, fingerprint feature extraction from a 1.5x1.5 mm image takes about 2 seconds and feature matching with an enrolled template takes about 0.03 seconds, with acceptable FRR (False Rejection Rate) and FAR (False Acceptance Rate). An identification out of N fingers thus can be carried out on average in $(2+0.03*N/2)$ seconds.

4.2. Sample applications

As an example application for mobile authentication, I have implemented a "pocket encryption unit." For encryption of data on a PC, the user puts his finger on the unit, which then generates a random key associated with the fingerprint and sends it to the PC via IrDA. The PC service program receives the message and uses the key to encrypt the data. Decryption is possible only when the user's finger is presented to the same unit, at which time a decryption key is sent from the unit to the PC. The IrDA communication itself is encrypted, and total security is guaranteed.

Another FpUI application I implemented is "fingertip remote browsing." By pointing the unit at a PC and changing fingers, the unit can be used to remotely control the browser on the PC by sending predefined URLs, and a user can enjoy easy net-surfing. Such a system of Web bookmarks indexed by fingers would be usable on any PC provided that the PC is running appropriate service software. Thus a user could metaphorically carry his favorite set of Web bookmarks in his Pocket-PID.

My evaluation of the implemented example applications on the prototype has confirmed that FpUI represents a significantly improved interface for palm-size PCs.

5. Conclusion

An enhanced user interface that takes advantage of a fingerprint identification technology, "FpUI," has been proposed and discussed. A mobile terminal with FpUI offers users both security with respect to fingerprint data and ease of use, a convenient man-machine interface utilized with just the touch of fingers.

FpUI is expected to be of increasing significance in the future when ever greater numbers of computer non-experts daily use digital appliances and intelligent mobile terminals for such Internet services as net-banking and electronic commerce. We hope to create a Pocket-PID, now the size of a palm-size PC, in the form of a card-size unit and also to implement it on such portable net-terminals as cellular phones and watches, so as to be literally pockettable.

Acknowledgments

I would like to thank Masanori Mizoguchi, Atsushi Sugiura and Jun Tsukumo for their very valuable comments and advice.

6. References

[1] L. O'Gorman, "Fingerprint Verification," in *Biometrics: Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999, pp. 43-64.

[2] A. Sugiura and Y. Koseki, "A User Interface Using Fingerprint Recognition," *UIST98*, San Francisco, 1998.

[3] Maybury, "Intelligent Multimedia Interfaces," *AAAI-Press/MIT-Press*, 1993.

[4] Rekimoto, "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments," *UIST97*, pp.31-39.

[5] K. Asai, et al, "Automatic Fingerprint Identification," *Proc. Society of Photo-Optical Instrumentation Engineers*, 182, 1979, 49-56..

[6] Ingris, et al, "A Robust, 1.8V, 250μW, Direct Contact 500 dpi Fingerprint Sensor," *ISSCC98, SA 17.7*, 1998, p. 285.